

Remarks

This reply is responsive to the Office communication mailed March 16, 2005.

The specification has been amended in the paragraph bridging pages 11 and 12 to correct a transposition error as well as certain grammatical errors.

Claims 1 and 15 have been amended to eliminate the recitation that at least one element of the first set has a context dependent relation to one or more elements of the second set. This change was made since at least one of the special case mappings described (from lowercase German "ß" to uppercase "SS") maps a single first element to plural second elements rather than being context dependent as such. Claims 1 and 15 have also been amended to recite that the first table contains an entry for each of the first set of elements. This is believed to have been implicit in the claims as originally presented, but is now made explicit.

New claims 17 and 18, dependent on claims 1 and 15, respectively, recite that at least one element of the first set has a context dependent relation to one or more elements of the second set and that the second table specifies, for each of a plurality of contexts, an element of the second set to be used in such context. This is shown, for example, in table 506 of Fig. 5, where source character x03A3 (Greek "Σ") is mapped to target character x03C2 ("ς") or x03C3 ("σ"), depending on whether the character is word final.

New claim 19 is based upon claim 1, but does not explicitly recite the steps involved in the creation of the first and second tables. More particularly, claim 19 is directed to a method for converting a first block of data comprising a first set of elements into a second block of data comprising a second set of elements in a computer system having a translation function. As further recited in the preamble, the translation function translates a block of elements of a first set into a block of elements of a second set in accordance with a translation table specifying for each element of the first set either one particular element of the second set or an exception handling element; the translation function is interrupted and an exception handling function is executed whenever an element being processed is marked by an exception handling element in

the translation table. It is also assumed that the first set of elements comprises a first subset consisting of such elements that are translated to one particular element of the second set and a second subset consisting of the remaining elements of the first set.

As further recited in claim 19, a first table is provided containing an entry for each of the first set of elements, with each element belonging to the first subset being assigned to the respective element of the second set and all elements of the second subset are assigned to an exception handling element. A second table is also provided specifying rules for translating the elements of the second subset. The first block of data is processed with the translation function, using the first table as a translation table. Upon encountering an element marked by an exception handling element in the first table, the exception handling function is executed to translate the element in accordance with rules specified by the second table.

New claim 20 is similar to claims 17 and 18, but depends on claim 19.

New claim 21, dependent on claim 19, recites that the translation function is implemented by a machine instruction of the computer system (page 5, line 21).

New claim 22 is a program product claim that is similar to claim 10, but depends on claim 19.

Finally, new claim 23 is similar to claim 19, but is directed to apparatus.

New claims 17-23, as well as claim 1-16 as amended, are believed to distinguish patentably over the prior art and, in particular, are believed to distinguish patentably over U.S. Patent 5,787,452 to McKenna ("McKenna"). McKenna discloses a system 200 (Fig. 2) in which a conversion engine 240 converts locale-specific textual data to Unicode, which is then handled by transformation processes 250 that include case mapping, among others. As shown more particularly in Fig. 7, case conversion is performed using at least a two-stage table lookup, in which the first byte of the Unicode character is used as an index into a first table 701 (implemented by a trie structure), while the second byte of the Unicode character is used as an index into a second table 711 (likewise implemented by a trie structure) referenced by the entry

703 in the first table 701. Each entry 713 in the second table contains either a final (two-byte) output value or a pointer to an entry 723 in a conversion heap 721, which is used for output values of nonstandard length. In the particular example shown in Fig. 7, conversion heap 721 is used to map a lowercase German "ß" to an uppercase "SS".

While the McKenna system obviously shares some similarities with applicant's invention, applicant's claimed invention differs in several key respects.

In the first place, applicant uses a translation function that translates a block of elements of a first set into a block of elements of a second set in accordance with a translation table. McKenna's translation function (Fig. 5), on the other hand, works on single characters and must be invoked iteratively to translate a string of characters.

Secondly, in applicant's claimed system, the first table contains an entry for each of the first set of elements. This means that for most first elements—which are mapped to particular second elements and not to a concatenation of second elements or to one of a plurality of elements depending on context—only a one-stage lookup in a single table is required. This is particularly advantageous where (as recited in claim 21) the machine on which the translation is performed has a hardware instruction (such as the TRTT instruction described in the specification) for performing the translation using such a translation table. By contrast, in the McKenna system, even discounting the initial conversion from a locale-specific text to Unicode, at least two table lookups (e.g., tables 701 and 711) are required, even for unexceptional cases not requiring resort to the conversion heap 721.

Thirdly, applicant executes an exception handling function to translate an element in accordance with rules specified (or represented) by the second table. While the Examiner has pointed to McKenna's conversion heap 721 as corresponding to applicant's claimed second table, there is no "function" being executed here, nor does the conversion heap specify "rules". Rather, the only processing that is different here is the additional step (507 in Fig. 5) of lookup in the conversion heap 721. That McKenna is incapable of translating in accordance with specified rules is evident when one looks at how he handles the other two cases mentioned by applicant:

one where the translation is dependent on position (final “ζ” versus a nonfinal “ο” corresponding to uppercase “Σ” in Greek) and the other where the translation is dependent on language (lowercase “ı” and “i” corresponding respectively to uppercase “İ” and “I” in Turkish). McKenna’s conversion heap 721 is plainly incapable of handling either of these situations, since it contains only a single mapping for each index value, regardless of position, language or any other context data.

While McKenna does consider locale-specific translations—e.g., standard French versus Canadian French uppercasing of lowercase accented characters (col. 9, lines 35-42)—he does not use his conversion heap 721 for this task. Rather, McKenna uses different trie structures for different locales (col. 9, lines 55-65). This greatly increases storage requirements, since the trie structures he uses for different locales are virtually identical to one another, differing only in the exception cases noted above. This contrasts with applicant’s claimed invention, which requires only a single table for one-to-one mappings while using a second table for exceptional cases.

For all of these reasons, new claims 19 and 23, as well as claims 1 and 15 as amended, and the claims dependent thereon are believed to distinguish patentably over the art cited by the Examiner.

New claims 17, 18 and 20 are additionally believed to distinguish patentably over the art cited by virtue of their recitation that at least one element of the first set has a context dependent relation to one or more elements of the second set and that the second table specifies, for each of a plurality of contexts, an element of the second set to be used in such context. By contrast, McKenna’s conversion heap 721 can only specify mappings independently of context.

New claim 21 is additionally believed to distinguish patentably over the art cited by virtue of its recitation that the translation function is implemented by a machine instruction of the computer system. By contrast, McKenna discloses a purely software implementation that as described is incapable of taking advantage of hardware assists.

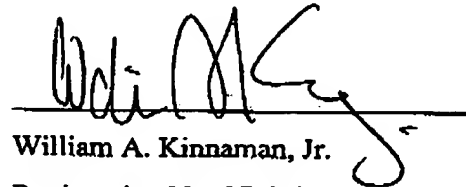
Conclusion

For the foregoing reasons, new claims 17-23 as well as claims 1-16 as amended are believed to distinguish patentably over the art cited by the Examiner.

Reconsideration of the application as amended is respectfully requested. It is hoped that upon such consideration, the Examiner will hold all claims allowable and pass the case to issue at an early date. Such action is earnestly solicited.

Respectfully submitted,
JOACHIM MANFRED BAUER

By



William A. Kinnaman, Jr.

Registration No. 27,650

Phone: (845) 433-1175

Fax: (845) 432-9601

WAK/wak